

1

{

File: minimal.pas

Compiler : Borland Pascal 7.0

Задача. На прямой ветке железной дороги расположено несколько станций. Задана стоимость проезда между любыми двумя станциями. Требуется написать программу нахождения минимальной стоимости проезда между любыми двумя станциями. Двигаться по железной дороге можно только в одном направлении (от станции с меньшим номером к станции с большим).

Входной файл Input.txt содержит в первой строке натуральное число N , не большее 250. Всего на дороге расположено $N + 1$ станций, пронумерованных от 0 до N . В следующих строках записано $N(N + 1)/2$ чисел, задающих стоимости проезда между станциями: сначала стоимость проезда от станции 0 до станций 1, 2, 3, ..., N , затем от станции 1 до 2, 3, ..., N ... от станции $N - 1$ до станции N . Все стоимости проезда – натуральные числа, не превосходящие 10000.

Выходной файл Output.txt должен содержать одно число – минимальную стоимость проезда от станции 0 до станции N с возможными пересадками.

| Input.txt | Output.txt |
|-----------|------------|
| 3 | 12 |
| 7 10 20 | |
| 4 8 | |
| 2 | |

Пояснение: В приведенном примере всего 4 станции с номерами 0, 1, 2, 3. Оптимальный маршрут проходит через станции с номерами 0, 2 и 3. Его стоимость равна $10 + 2 = 12$.

Решение. Калмыков Вадим (ProCrypt),
г. Сургут, ЦНИТ "Северная Звезда",
11:06, 10.07.2008

Source : <http://acm.dvpion.ru>

Editor. MSP, 15.12.09, 10:12; LIST 6.0 (c) 1999–2009

}

39 {\$R-}

40 Program Minimal;

41 Type

```

42     ArrType = Array [1..1] of Word; { массив для одного элемента }
43
44     Var
45     i, j : byte; { счетчики циклов }
46
47     A : ^ArrType; { указатель на тип ArrType }
48
49     {
50     A [i] - минимальная стоимость проезда до i-ой станции
51     }
52
53     d : Word; { стоимость проезда от станции i до станции j }
54
55     N : byte; { номер последней станции }
56
57     Begin
58     Assign (Input, 'input.txt');
59     Assign (Output, 'output.txt');
60
61     Reset (Input);
62     ReadLn (N);
63
64     { Выделяем динамическую память под массив }
65     GetMem (A, SizeOf (ArrType) * N);
66
67     { заполняем массив первой строкой входного файла }
68     for i := 1 to N do
69     Read (A^ [i]);
70
71     for i := 1 to N do
72     for j := (i + 1) to N do
73     begin
74     Read (d);
75
76     { если выгодно, то совершаем пересадку }
77     if (A^ [i] + d < A^ [j])
78     then
79     A^ [j] := A^ [i] + d;
80     end;
81
82     Close (Input);
83
84     ReWrite (Output);
85     Write (A^ [N]);
86
87     Close (Output);
88
89     { возвращаем выделенную память под управление ОС }
90     FreeMem (A, SizeOf (ArrType) * N);
91     End.

```

Listing данной задачи опубликован в сети Internet по адресу

<http://www.Best-Listing.ru/color-10-task-669.html>

Sergey Mitrofanov, 19.09.14, 20:42

E-mail: infostar@mail.ru

© <http://www.Best-Listing.ru/>, 2006–2014